Random Forests: Why They Work and Why That's a Problem

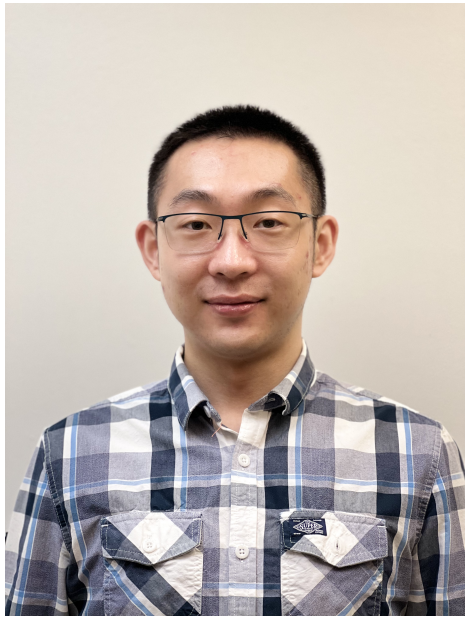
AI, Statistics, and Data Science in Practice
NISS Collaboratory

Lucas Mentch
Associate Professor
Dept. of Statistics
University of Pittsburgh

November 19, 2024

Acknowledgements

- Much of what follows is joint work with a former graduate student (now postdoc at Emory): Siyu Zhou



- Work within sponsored by the National Science Foundation

Overview

1. Setup and Foreshadowing: Original Solutions to Old Problems
   - Trees and random forests
   - Early work connecting random forests and u-statistics
   - Asymptotic normality and hypothesis tests for feature importance
   - [Mentch and Hooker, 2016, 2017, Coleman et al., 2022]

2. Why do random forests work?
   - Randomization as regularization
   - Degrees of freedom
   - Exporting the RF mechanism

3. Why that's a problem ... new perspectives bring new problems
   - Alternative forms of regularization
   - Implications for VIMP measures

Part I: Introduction

The Random Forest (RF) procedure is a supervised learning tool introduced by Breiman [2001].

Assume we have data of the form $\mathcal{D}_n = \{Z_1, ..., Z_n\}$ where each $Z_i = (X_i, Y_i)$, $X_i = (X_{1,i}, ..., X_{p,i})$ denotes a vector of p features, $Y \in \mathbb{R}$ denotes the response, and the variables have a general relationship of the form $Y = f(X) + \epsilon$.

Random forests are constructed by drawing B resamples of the original data, building a (randomized) model (base-learner) on each, and taking the average. Given some point x, the RF prediction is given by

$$\hat{y} = \text{RF}(x; \mathcal{D}_n, \Theta) = \frac{1}{B} \sum_{i=1}^{B} T(x; \mathcal{D}_n, \Theta_i)$$

Introduction

In the original construction of RF given by Breiman [2001] ...

- Resampling is done via bootstrapping
- Base learners are CART-style trees
- Trees are recommended to be fully-grown
- Besides drawing resamples, the randomness Θ serves to randomly select a subset of $\texttt{mtry} < p$ features available for splitting at each node of each tree
- For classification problems, split quality measured via reduction in Gini impurity and final predictions taken as majority vote

Note: Many recent works (including mine) take a more general RF definition for theoretical convenience

Subsampling

- Goal of our early work was to develop inferential procedures for RFs – greedy nature and bootstrapping (correlation) complicate this
- Instead let's do **subsampling** – construct trees with m_n *subsamples* of size k_n instead of full bootstrap samples

$$\hat{f}(x) = \frac{1}{m_n} \sum_{i=1}^{m_n} T(x; (\boldsymbol{X}, Y)_{i_1}, ..., (\boldsymbol{X}, Y)_{i_{k_n}}; \Theta)$$

- Can draw connection to u-statistics (randomized kernels with growing rank) $\implies$ predictions are asymptotically normal so long as subsample growth rate is controlled
- Confidence intervals can be produced with consistent variance estimate; rates of convergence (Berry-Esseen Theorems) can be established [Peng et al., 2022]

Testing Feature Significance

- Asymptotic results also provide a means of formally testing feature importance (details given on next slide):
  - Given $S \subset \{X_1, ..., X_p\}$, build RF and RF* with permuted or randomized replacement for S
  - Predictions from each are AN $\implies$ differences should be AN and centered at 0 under H_0 $\implies$ large differences in predictions should indicate a contribution by the features in S
  - Can be extended to test additivity by structuring the test points [Mentch and Hooker, 2017]
  - Need to estimate the (co)variance (lots of trees); becomes very difficult for large numbers of test points
  - More efficient and scalable version of this test that involves exchanging trees between the two forests given in Coleman et al. [2022]

Real Data: Indigo Bunting Presence/Absence 2010

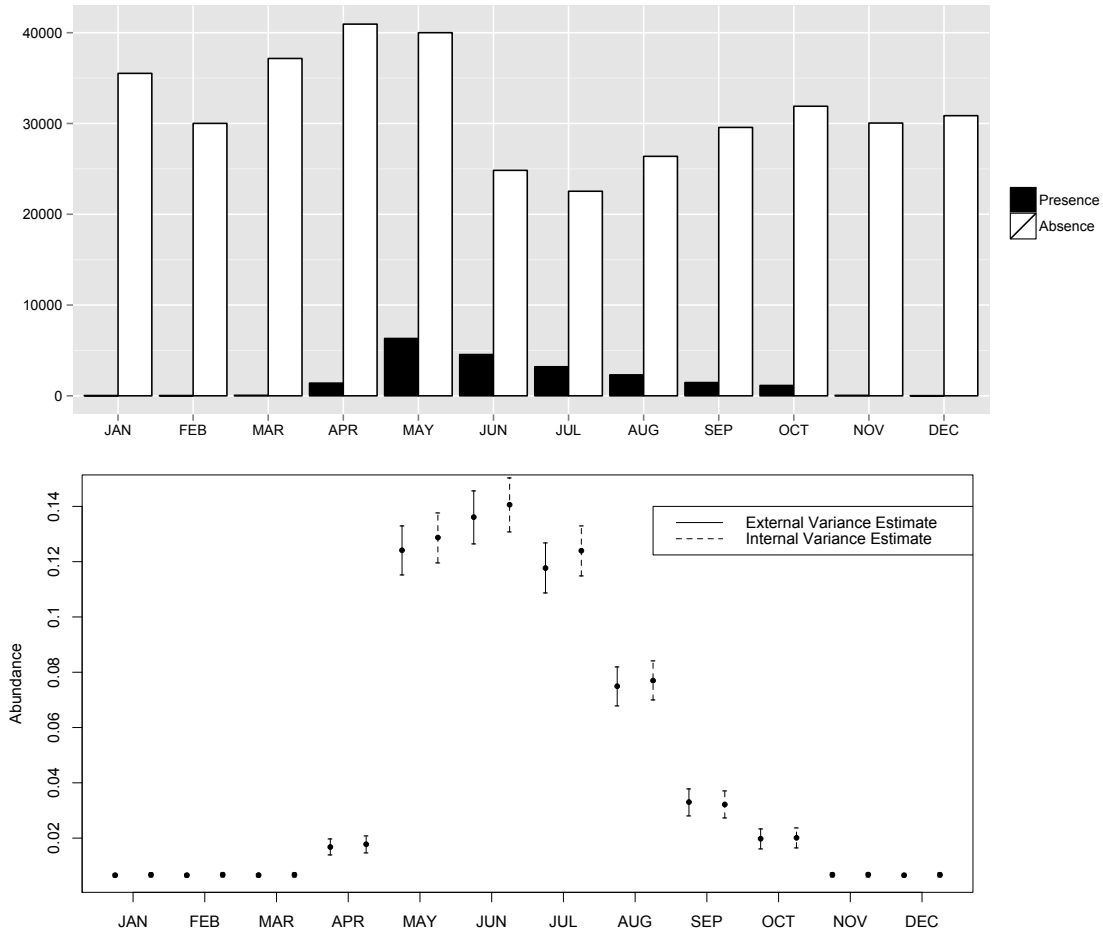


Figure 1: Monthly counts of Indigo Bunting observations in 2010.

Real Data: Indigo Bunting: Effect of Month

- Testing the importance of month during the year 2010. (Critical Value from $\chi^2_{20} = 31.41$)

Test	Test Statistic	p-value
Month vs. No Month	4233.10	0.0000
Randomized Month vs. No Month	58.02	0.0001
Randomized Month vs. Randomized Month	2.36	0.9999
Month vs. Randomized Month	2336.14	0.0000

- Final test implies that Month improves prediction beyond what could be expected by simply adding random noise.

Part II: Why do RFs Work?

Despite the vast recent theoretical/statistical progress with RFs, there has been very little work exploring reasons for their robust success

From recent review paper by Biau and Scornet [2016]:

- Research investigating the properties of random forest tuning parameters is "unfortunately rare"
- "present results are insufficient to explain in full generality the remarkable behavior of random forests."

Good reason to think there really is something particularly nice about RFs: recent large-scale empirical study comparing 100s of models on 100s of datasets (entire UCI repository), Fernández-Delgado et al. [2014] found:

- RFs ranked 1^{st} overall and of the top 5 performing classifiers, 3 were some variant of RFs

Existing Explanations

1. Breiman [2001]: The additional randomness in RFs serves to de-correlate trees, thereby reducing the variance of the ensemble (accuracy/correlation tradeoff $\iff$ bias/variance tradeoff)

2. Biau and Scornet [2016]: "*The authors' intuition is that tree aggregation models are able to estimate patterns that are more complex than classical ones—patterns that cannot be simply characterized by standard sparsity or smoothness conditions.*"

3. Wyner et al. [2017]: Random forests work well because they are "self-averaging interpolators" that fit the training data perfectly while retaining some degree of smoothness due to the averaging – Random forests "*work not in spite, but because of interpolation*"

Relative Performance of RFs

Perhaps it would be more fruitful to consider RF performance in **relative** terms (why do RFs do well compared to other models) instead of **absolute** terms (why *might* models *like* RFs do well in general)

Let's look at some preliminary experiments ...

- Linear model with correlated features (following Hastie et al. [2017])
- MARS model (interactions and non-linearities; Friedman [1991])
- Consider SNRs ranging from 0.05 to 6 equally spaced on the log scale
- Compare difference in test error of random forests ($\texttt{mtry} = 0.33$) to bagging ($\texttt{mtry} = 1$) averaged over $N = 500$ simulations. **Note:** Slightly different convention here – `mtry` denotes the *proportion* of the p features available for splitting, rather than the raw number

Relative Performance of RFs

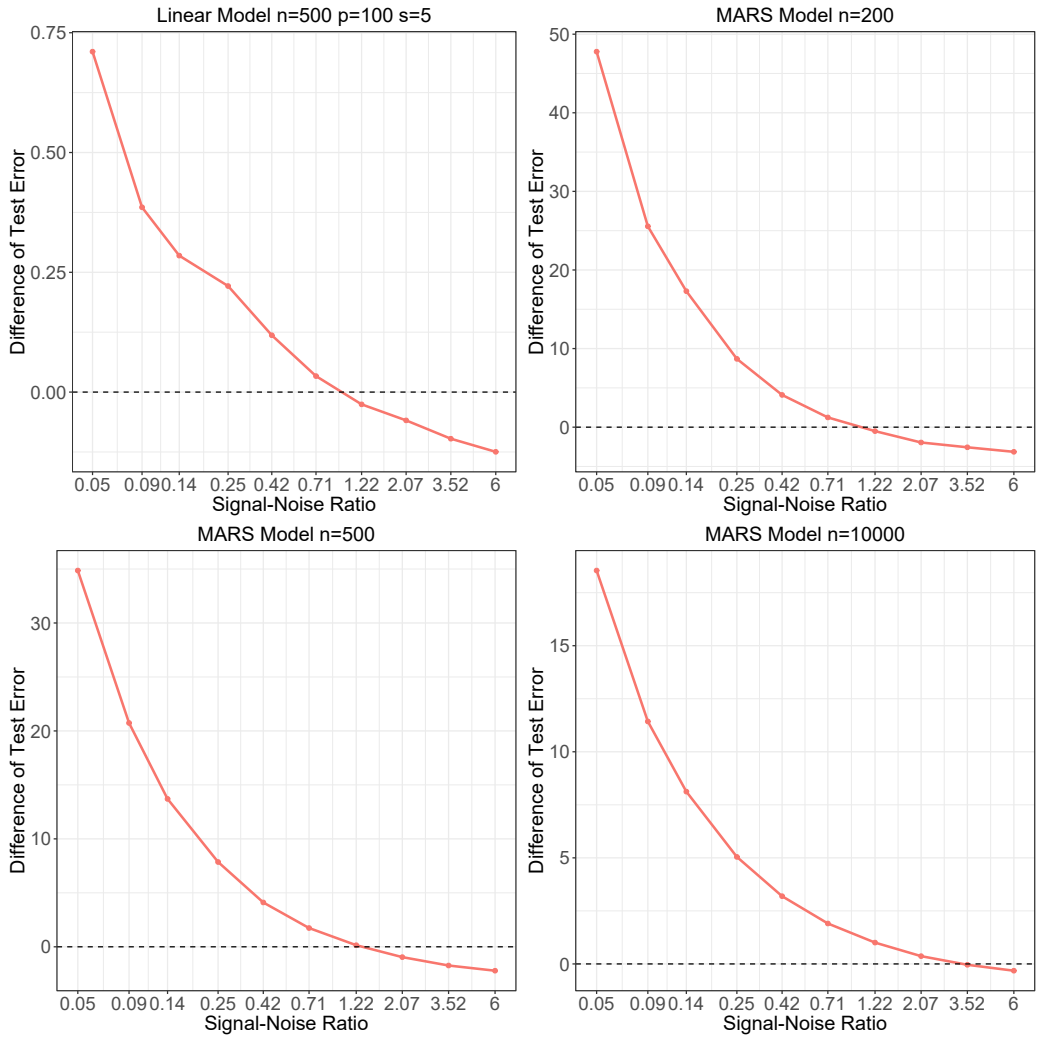


Figure 2: Error(Bagg) - Error(RF) vs SNR. Positive values indicate better performance by RFs.

Relative Performance of RFs

In each case we see a clear pattern: as the SNR goes up, the advantage offered by RFs dies out.

How about on real-world data?

- Since we don't know the true SNRs, we inject additional random noise $\epsilon \sim N(0, \sigma^2)$ into the response
- σ^2 chosen as a proportion α of the sample variance of the response for $\alpha = 0, 0.01, 0.05, 0.1, 0.25, 0.5$
- Consider the relative test error defined by

$$\text{RTE} = \frac{\widehat{Err}(\text{Bagg}) - \widehat{Err}(\text{RF})}{\hat{\sigma}_y^2} \times 100\%$$

where $\widehat{Err}(\text{Bagg})$ and $\widehat{Err}(\text{RF})$ correspond to 10-fold CV error and $\hat{\sigma}_y^2$ is the empirical variance of the original response.

Relative Performance of RFs

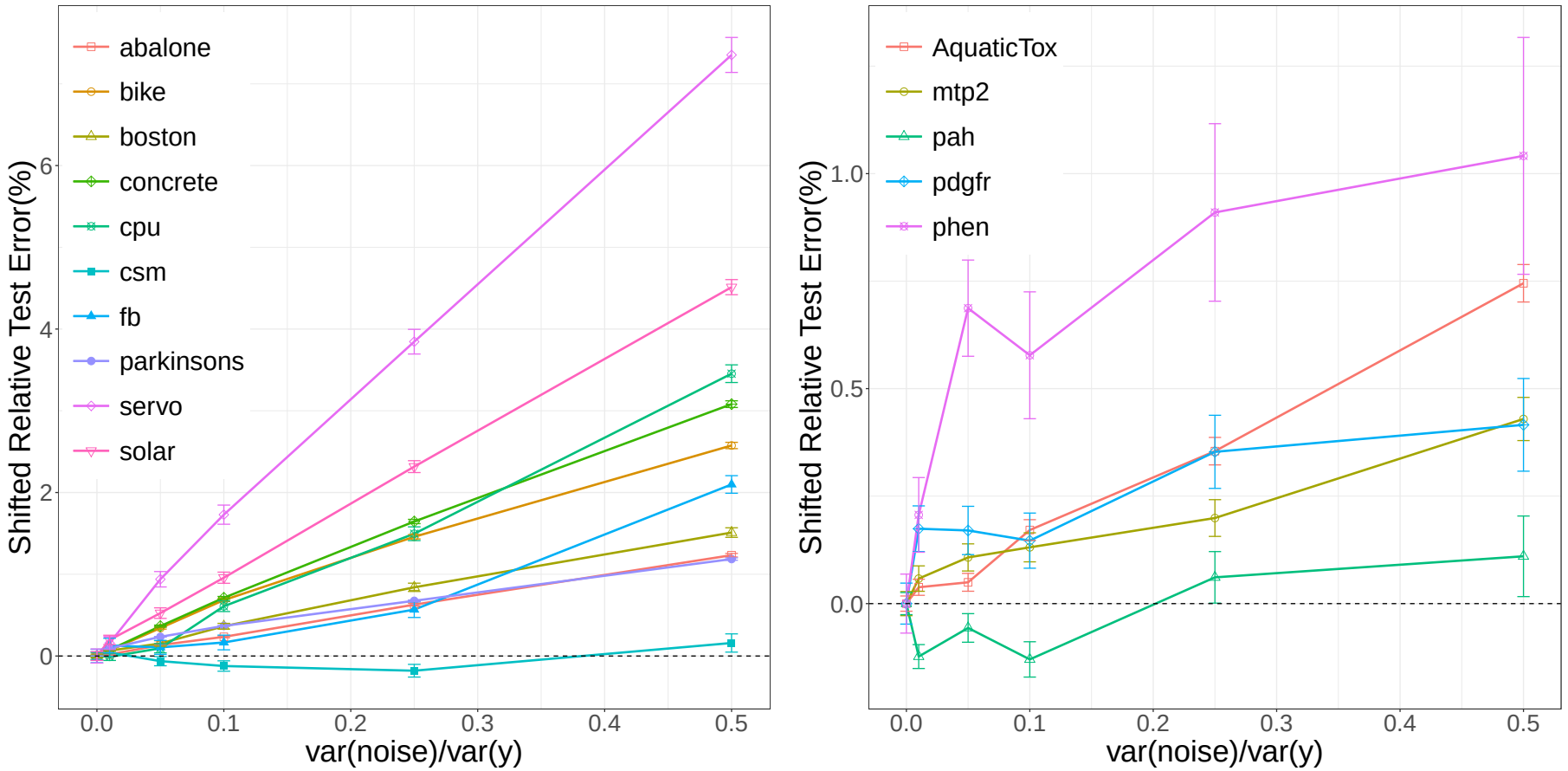


Figure 3: (Shifted) RTE on real data where additional noise is added. The left plot shows results on low-dimensional datasets taken from the UCI repository; the right plot shows results on high-dimensional datasets.

Takeaways

- Turns out, you can demonstrate that RFs with larger values of `mtry` (more variables at each split; less random) have more degrees of freedom
- Results are interesting and helpful, but perhaps not shocking: more randomness $\implies$ trees more independent $\implies$ bigger variance reduction $\implies$ less overfitting $\implies$ improved performance at low SNRs.
- More surprising: there is nothing tree-specific about this. Could we export the RF mechanism (feature subsampling) into other modeling/fitting procedures and see similar gains?
  - **BaggFS**: Bootstrap original data, perform linear model forward selection (FS) on each, take average
  - **RandFS**: Same as BaggFS, except that only random subset of remaining predictors available at each step in the FS procedure

Extensions to RandFS

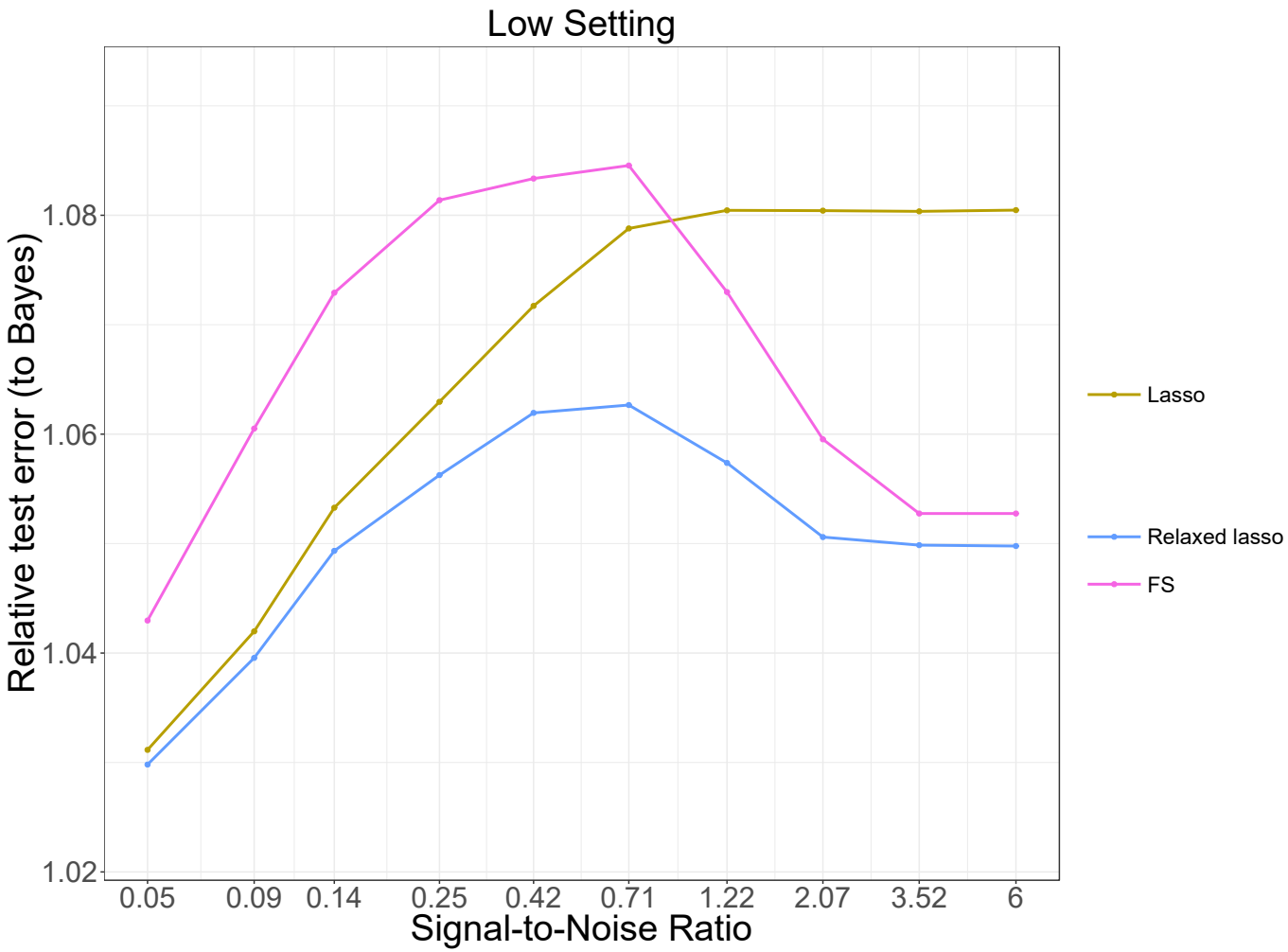


Figure 4: Performance Comparisons in low setting.

Extensions to RandFS

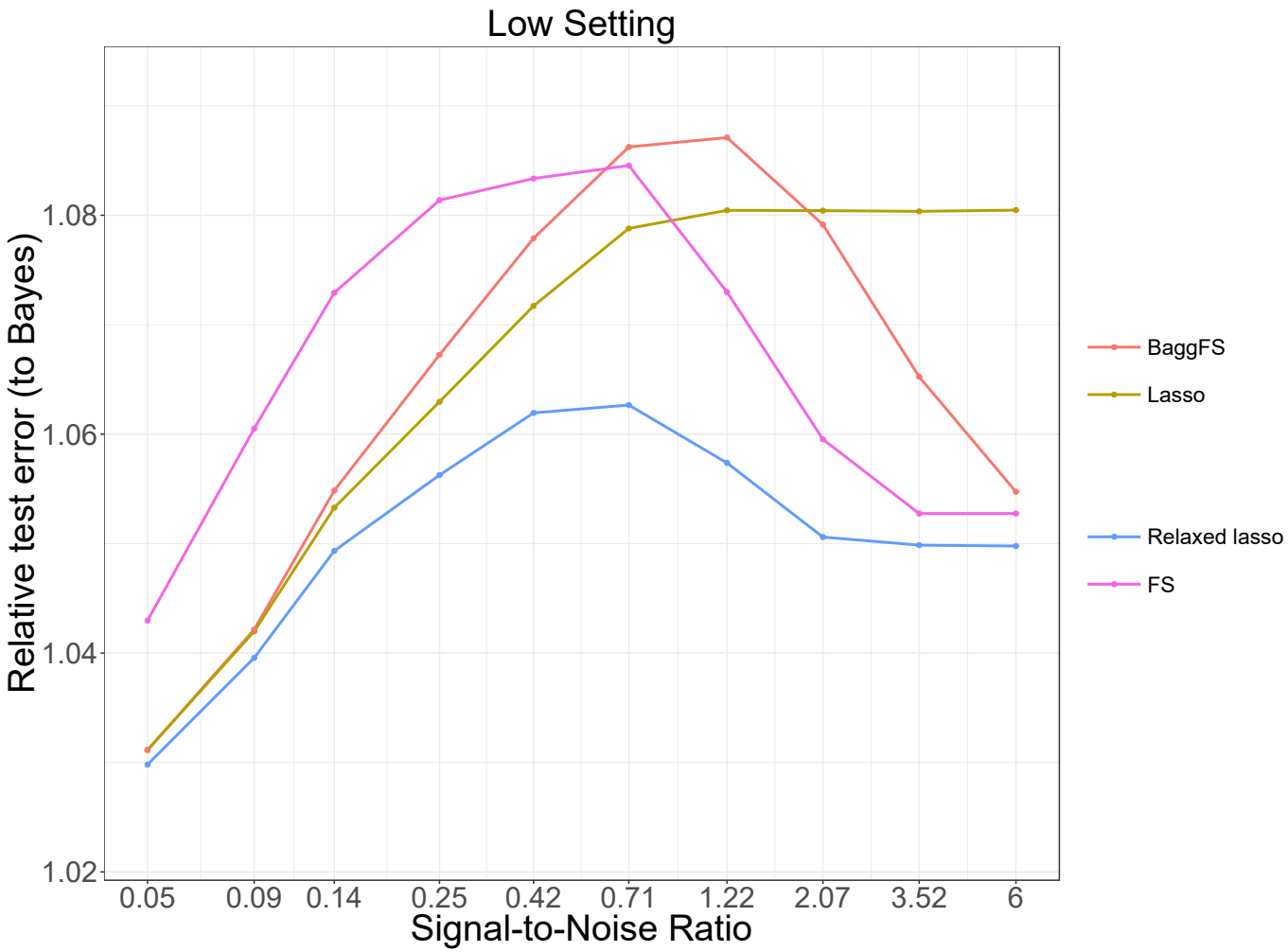


Figure 5: Performance Comparisons in low setting.

Extensions to RandFS

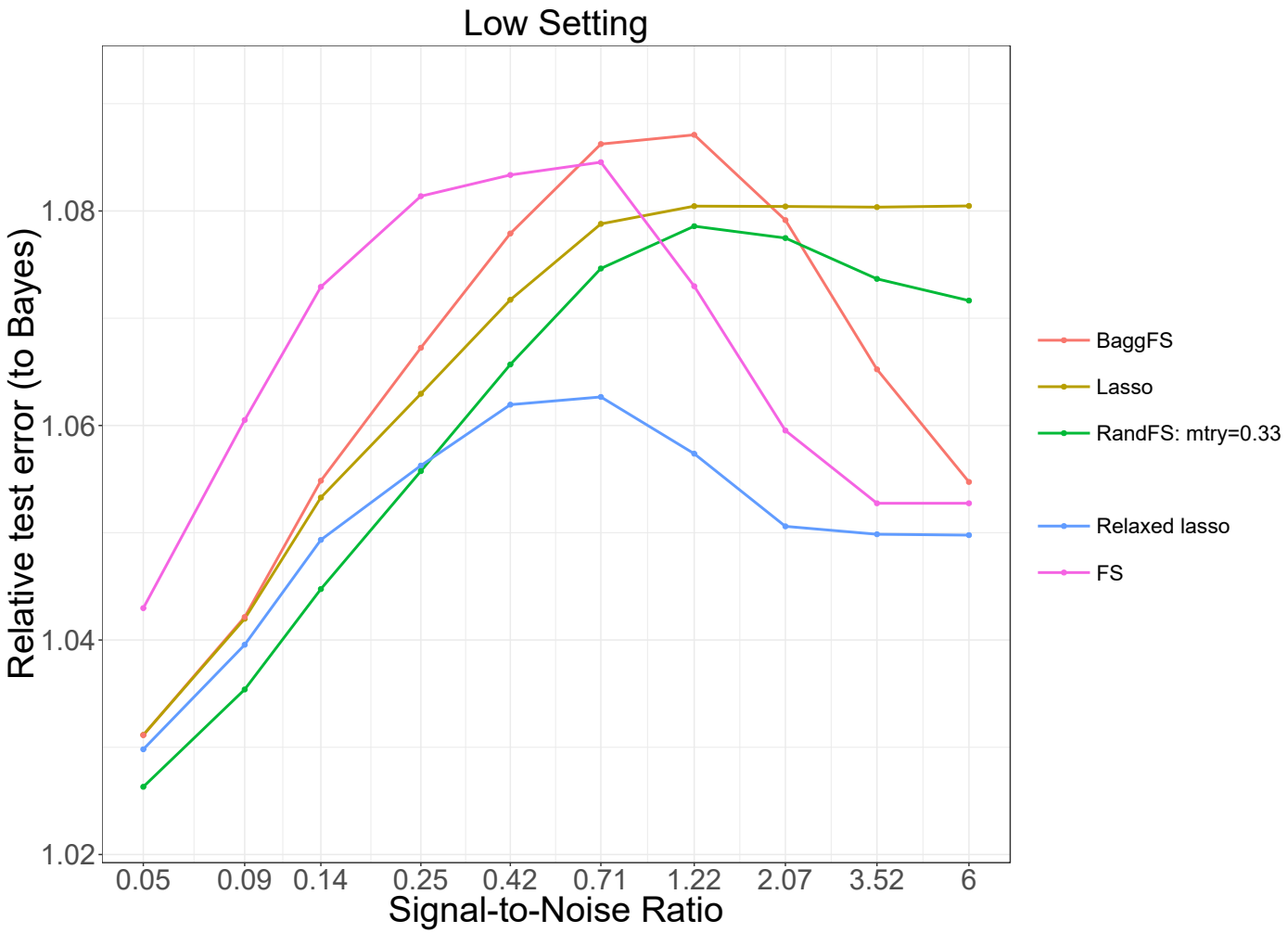


Figure 6: Performance Comparisons in low setting.

Extensions to RandFS

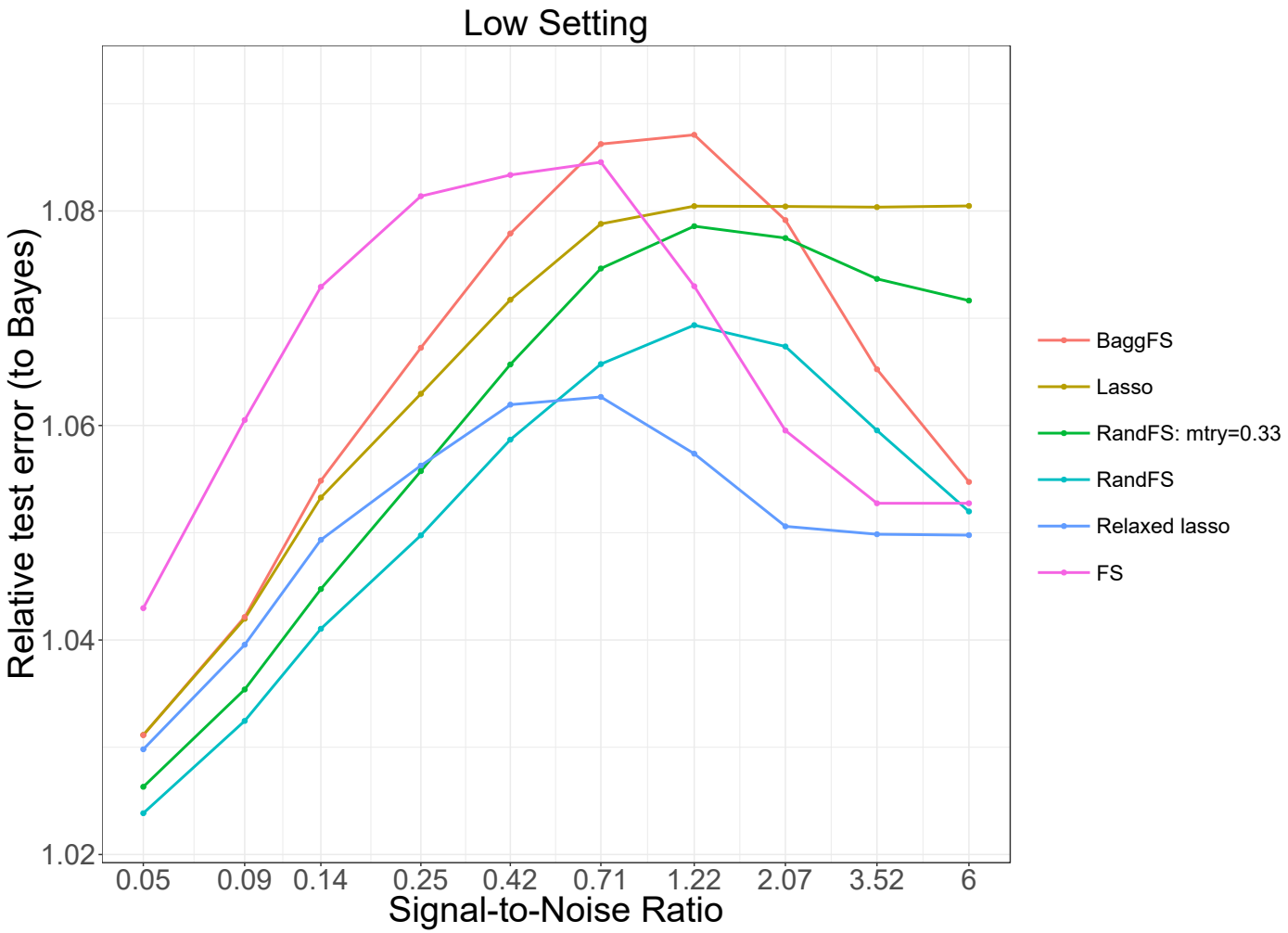


Figure 7: Performance Comparisons in low setting.

Extensions to RandFS

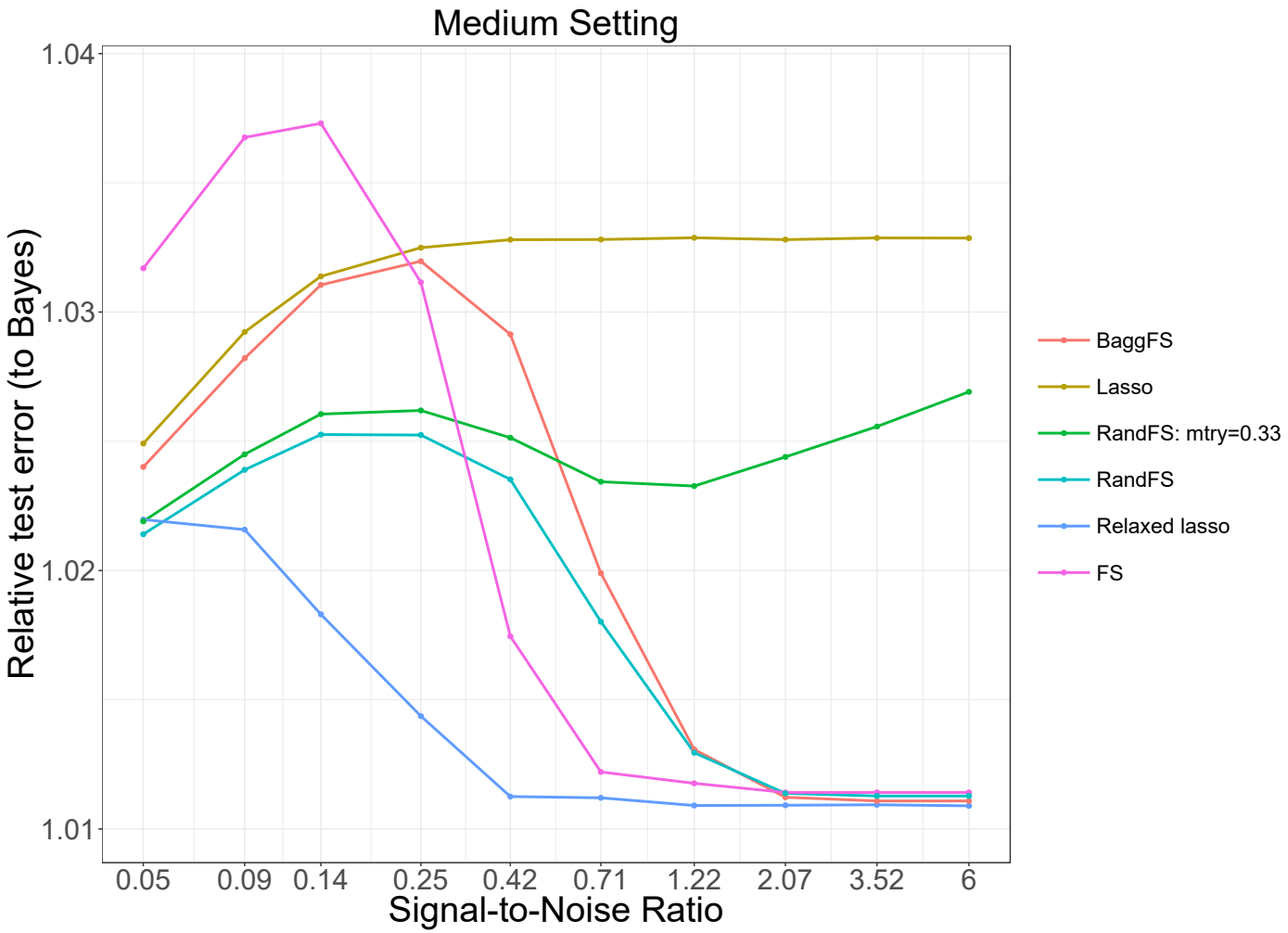


Figure 8: Performance Comparisons in medium setting.

Randomization as Regularization

Randomized forward selection is doing a sort of *implicit* regularization: imagine we do an idealized version of this kind of randomized LM forward selection without resampling to a "depth" of $d < p$

- For each of the $b = 1, ..., B$ models, features are either selected or not $\implies$ for each feature X_k, $\hat{\beta}_{k,b} = 0$ w.p. ≥ 0
- Given an orthogonal design:

$$\hat{\beta}_k^{\text{RandFS}} = \frac{1}{B} \sum_k \hat{\beta}_{k,b} = \alpha_k \hat{\beta}_{k,OLS} + (1 - \alpha_k) \cdot 0 = \alpha_k \hat{\beta}_{k,OLS}$$

- Each selection proportion/shrinkage α_k depends on:

  (i) the probability X_k is made eligable (`mtry` and model depth d) and (ii) the probability of being chosen if eligible (true relative "importance")

Randomization as Regularization

- Coefficient estimates are effectively shrunk by amount proportional to that selection proportion
- In fact, can show that RandFS with `mtry` parameter is equivalent to ridge regression with penalty $\lambda = \frac{p - \texttt{mtry}}{\texttt{mtry}}$:

Theorem (Mentch & Zhou (2020))

Under the data setup given above, assume that $n > p$ and the design matrix $\boldsymbol{X}$ is orthogonal. Then

$$\hat{\beta}^{ens} \xrightarrow{B \to \infty} \frac{mtry}{p} \hat{\beta}^{OLS}$$

where $\hat{\beta}^{ens}$ denotes the estimate formed by averaging across B different OLS models, each built using only a subset of $mtry < p$ features selected uniformly at random, and $\hat{\beta}^{OLS}$ denotes the standard OLS estimate on the original data.

Part III: The "Problem"

- We just argued that the randomization (`mtry` parameter) in RFs acts as a regularizer – each feature is effectively shrunk by an amount proportional to its selection proportion α_k
- **Key Point:** Column subsampling (`mtry`) is only one way to affect those selection proportions – if we hold `mtry` fixed and add more features, shouldn't we intensify the effect?
  - In other words it's not really `mtry` alone at work, but really `mtry`$/p$ and we just think of p as fixed
- If that's really what is "making RFs work", then bagging with extra features should improve predictions in a similar fashion to RFs

Augmented Bagging

Suppose we create an augmented dataset $\mathcal{D}_n^* = \{\boldsymbol{Z}_1^*, ..., \boldsymbol{Z}_n^*\}$ where each $\boldsymbol{Z}_i^* = (\boldsymbol{X}_i, \boldsymbol{N}_i, Y_i)$ and $\boldsymbol{N}_i = (N_{1,i}, ..., N_{q,i})$ is an additional set of noise features. The original bagging procedure is then performed on $\mathcal{D}_n^*$ so that these predictions from *Augmented Bagging* (AugBagg) take the form

$$\hat{y}_{\text{AugBagg}} = \frac{1}{B}\sum_{b=1}^{B} T((\boldsymbol{x}, \boldsymbol{n}); \Theta_b, \mathcal{D}_n^*). \tag{1}$$

where $\boldsymbol{n}$ can be filled in with random draws from the additional noise features.

- We assume only that $\boldsymbol{N}$ is conditionally independent of Y given $\boldsymbol{X}$.

Simulations on AugBagg

Consider the same general linear model setup as before:

- Set $n = 100$, $p = s = 5$.
- q additional i.i.d. noise features sampled from $\mathcal{N}(0, 1)$ independent of $\boldsymbol{X}$ are then added with q ranging from 1 to 250.
- The noise term ϵ was sampled from $\mathcal{N}(0, \sigma_\epsilon^2)$ with σ_ϵ^2 chosen to satisfy a particular SNR given by $\beta^T \Sigma \beta / \sigma_\epsilon^2$.
- SNR $= 0.01$, 0.05, 0.09, or 0.14.
- Models built via the R package `randomForest` with default settings except for `mtry` $= p + q$.
- Test error is calculated on an independent, randomly generated test set containing 1000 observations and averaged over 500 repetitions.

Simulations on AugBagg

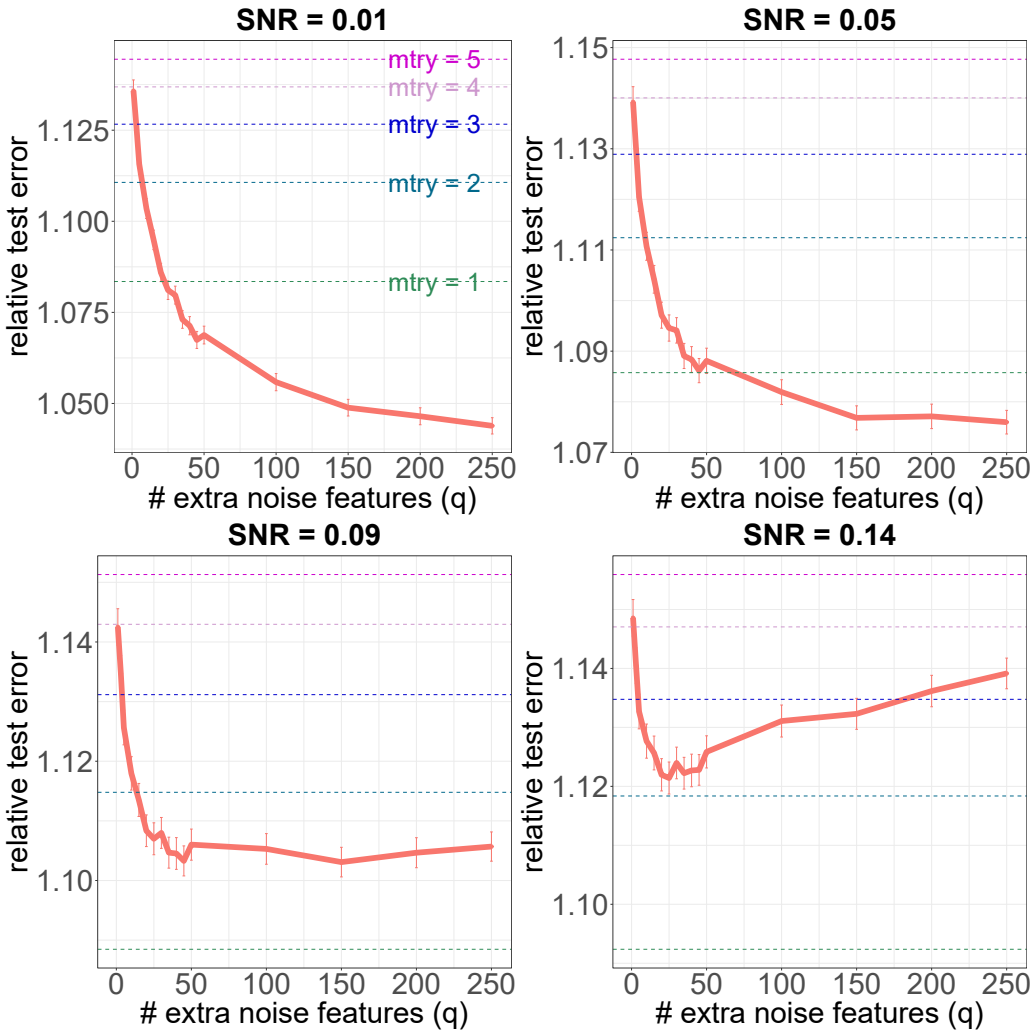


Figure 9: Performance of Augmented Bagging

Variable Importance

- The fact that inclusion of noise features can *potentially* improve model performance has a crucial implication for variable importance, which is of interest across almost all scientific domains.
- Given $Y = f(\boldsymbol{X}) + \epsilon$, many different versions of testing procedures developed that (generally speaking) proceed by partitioning $\boldsymbol{X} = (\boldsymbol{X}_0, \boldsymbol{X}_{\text{test}})$ and evaluating a null hypothesis along the lines of

$$H_0 : f(\boldsymbol{X}_0, \boldsymbol{X}_{\text{test}}) = f_0(\boldsymbol{X}_0) \tag{2}$$

  [Mentch and Hooker, 2016, 2017, Coleman et al., 2022, Lei et al., 2018]

- Generally involves building two models – one with the original features and one in which $\boldsymbol{X}_{\text{test}}$ is either **dropped** or **randomly replaced**

Variable Importance

But ...

- If model performance can be improved simply by adding randomly generated features that are (at least conditionally) independent of the response, then observing a significant drop in accuracy when a particular set of features is excluded does not imply that any relationship to the response or even the other covariates need exist.
- To emphasize this point, we implement recent testing procedures and investigate their behavior under the same linear model settings as above.
  - Consider $\boldsymbol{X}_0 = (X_1, ..., X_5)$ (true signal features) and $\boldsymbol{X}_{\text{test}} = (N_1, ..., N_q)$ (noise features)
  - $\boldsymbol{X}_0$ and $\boldsymbol{X}_{\text{test}}$ are independent or correlated.
  - In the altered dataset, $\boldsymbol{X}_{\text{test}}$ is dropped (drop test) or replaced (replacement test)

Tests: Independent Features

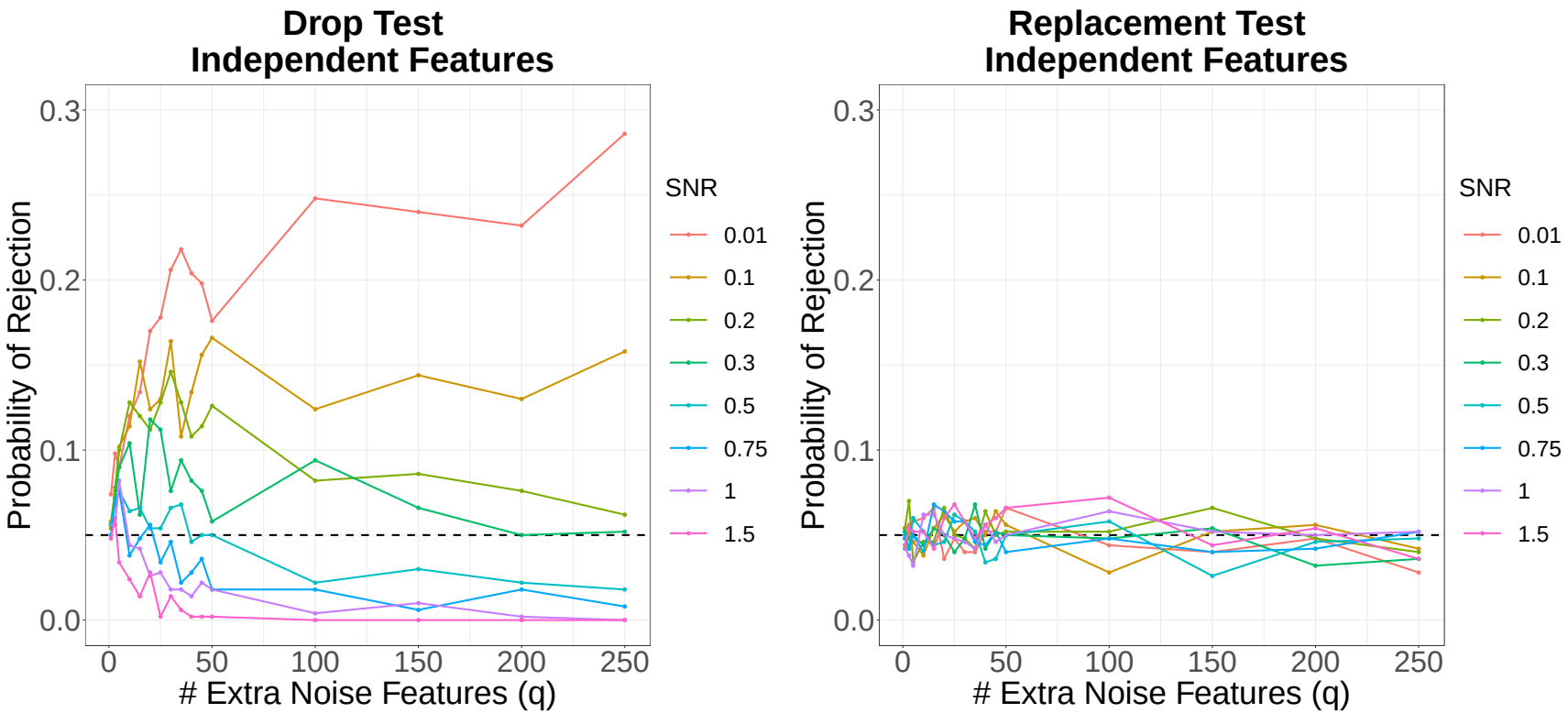


Figure 10: Probability of Rejecting Null Hypothesis when dropping vs replacing independent features.

Tests: Correlated Features

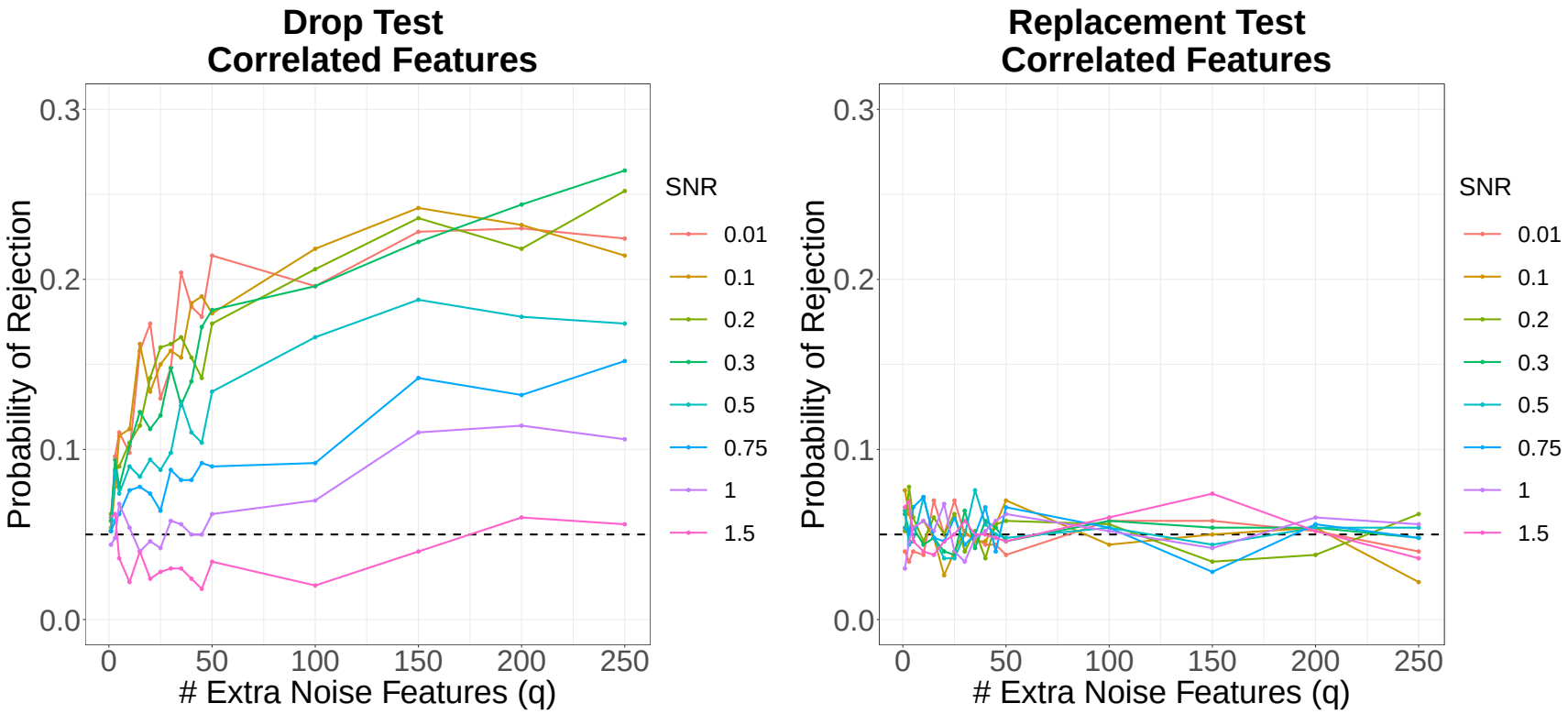


Figure 11: Probability of Rejecting Null Hypothesis when dropping vs replacing correlated features.

Tests: Features Sampled from Wrong Distribution

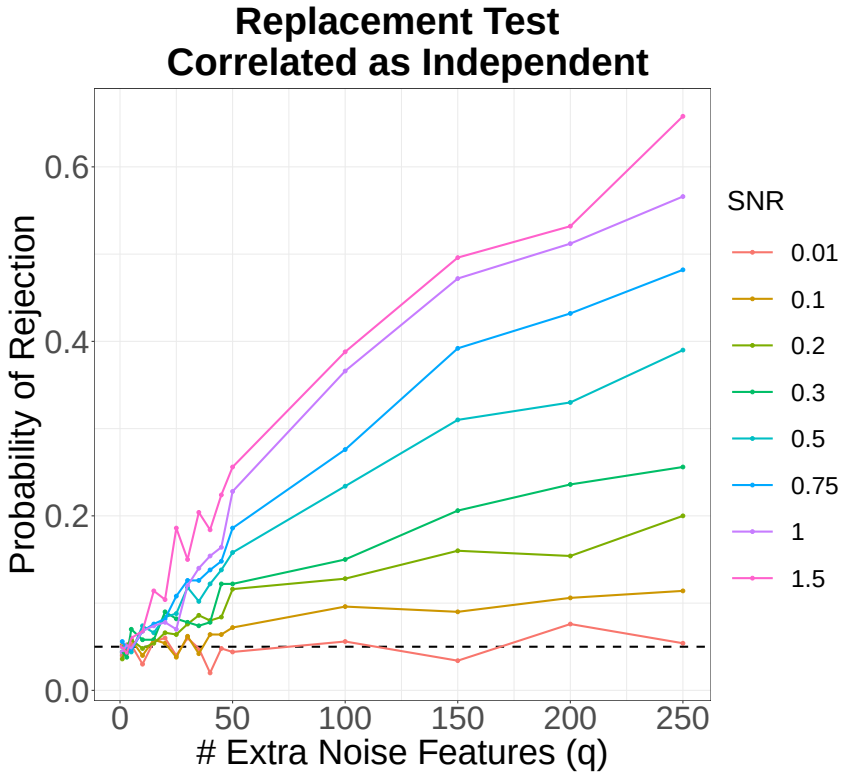


Figure 12: Probability of Rejecting Null Hypothesis when replacing features with others from a different distribution.

Bad Tests or Bad Interpretations?

So are these tests just wrong ... ?

Well not exactly ... the tests themselves are simply looking for a difference in predictions/accuracies between the models. Adding noise really *can* improve the models.

But there is a big issue with equating

"feature set S improves predictions"

with

"feature set S must explain/be related to the response"

Takeaways

- Additional randomness in RFs offers an implicit form of regularization, making them particular strong in noisy data settings
    - Just as ridge/lasso are thought of as regularized versions of OLS, can think of RFs as regularized form of bagging
- This kind of regularization can also be accomplished by adding additional noise features
    - Tests that measure importance as a function of drop in model accuracy when features are removed can produce highly misleading results
    - Still preferable to traditional out-of-bag measures which have been shown to suffer serious systematic bias
    - Effect can largely be mitigated by replacing features in question with knockoffs rather than dropping them

Takeaways

- These are *not* tree-specific results – similar "shrinkage-via-additional noise" effects can be seen in things as simple as linear models
  - "Optimal ridge penalty can be 0 even in high dimensions" [Kobak et al., 2020]
- This may not be the whole story – at least in some settings, RFs can continue to improve over bagging even at very high SNRs
  - "Randomization Can Reduce Both Bias and Variance: A Case Study in Random Forests" [Liu and Mazumder, 2024]

Relevant Work

- **Part II: Why Random Forests Work:**

  Lucas Mentch and Siyu Zhou. (2020). "Randomization as regularization: A degrees of freedom explanation for random forest success." *Journal of Machine Learning Research*. 21(171).

- **Part III: Why That's a Problem:**

  Lucas Mentch and Siyu Zhou. (2022). "Getting Better from Worse: Augmented Bagging and A Cautionary Tale of Variable Importance." *Journal of Machine Learning Research*. 23(224).

References I

Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2): 197–227, 2016.

Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

Tim Coleman, Wei Peng, and Lucas Mentch. Scalable and efficient hypothesis testing with random forests. *The Journal of Machine Learning Research*, 23 (170):1–35, 2022.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Jerome H Friedman. Multivariate Adaptive Regression Splines. *The annals of statistics*, pages 1–67, 1991.

Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*, 2017.

References II

Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez. The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *J. Mach. Learn. Res.*, 21:169–1, 2020.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Brian Liu and Rahul Mazumder. Randomization can reduce both bias and variance: A case study in random forests. *arXiv preprint arXiv:2402.12668*, 2024.

Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(1):841–881, 2016.

Lucas Mentch and Giles Hooker. Formal hypothesis tests for additive structure in random forests. *Journal of Computational and Graphical Statistics*, 26(3): 589–597, 2017.

References III

Wei Peng, Tim Coleman, and Lucas Mentch. Rates of convergence for random forests via generalized u-statistics. *Electronic Journal of Statistics*, 16(1): 232–292, 2022.

Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.